Relatório Competição 1

Matheus Andrade Dutra e Mariana Inácia Xavier Borges

Dezembro 2022

1 Resumo

Este trabalho é baseado na análise de dados médicos de operadoras de seguros de saúde e tem como objetivo ajudar a automatizar o processo em termos de análise de requisitos. Sendo assim, foi feita uma análise dos dados, observando principalmente quais campos estavam vazios, a quantidade de dados presentes no documento, a quantidade de classes e classes que cada campo possuía.

O conjunto de dados de treinamento fornecido contém 227.122 dados e 32 variáveis, dentre elas o rótulo "aprovar ou negar" e o outro representa apenas o ID do usuário. Das outras 30 variáveis fornecidas, apenas 15 ou 50% não possuem campos vazio. Tempo de doença e sua unidade (dias, semanas, meses), tipo de doença 99% dos campos estão vazios e o tipo de consulta contém 95% dos campos Não usado. O tipo de saída também era completamente nulo.

2 Pré-processamentos realizados

- Pré processamento utilizados:

  Foi preenchido colunas que não continham dados em valores 0 ou string de '0', de forma que não existisse celulas vazias.

  StandardZation: Metodo utilizado pra mudar a escala dos seus dados, digamos de uma variavel numerica de 3 a 345 em valores, pra um alcance de 0 a 1.

  LabelEncoder: Utilizado em váriaveis categóricas, tem como objetivo codificar os labels de destino com valor entre 0 e nclasses-1.

  OneHotEncoder: Codificado as características categóricas como uma matriz numérica única.

3 Configuração experimental

Foram utilizadas as seguintes configurações: Parâmetros utilizados:

- Numericos:

  NR_SEQ_REQUISICAO,

  QT_SOLICITADA,

  QT_DIA_SOLICITADO

- Categoricos(booleanos):

  DS_CBO.

  DS_TIPO_INTERNACAO.

  DS_CARATER_ATENDIMENTO.

  DS_TIPO_ATENDIMENTO.

  DS_INDICACAO_ACIDENTE.

  DS_TIPO_PREST_SOLICITANTE.

  DS_CLASSE.

  DS_TIPO_GUIA.

  DS_SUBGRUPO.

  DS_GRUPO.

- Target:

  DS_STATUS_ITEM

4 Algoritmos utilizados

- Linguagem: Python 3.10;
- Bibliotecas: numpy, pandas, os, sklearn.preprocessing importado = StandardScaler, LabelEncoder, OneHotEncoder.
- Foi utilizado o algoritmo RandomForestClassifier, afinal por ter uma coleção de dados para apontar uma decisão, este algoritmo seria o mais preparado para a tarefa de classificar a acurácia da cobertura do plano de saúde.
- A biblioteca fornecida pelo SKLearn foi muito útil, pois nos permitiu compartilhar a implementação da função train-test-split, baseado em dados de treinamento e teste. Os parâmetros que foram usados nesta função foram: tamanho do teste = 0,3, 30% para teste, 70% para treinamento, aleatório = Verdadeiro, assim tentando melhorar a distribuição de aleatoriedade nos dados.

	precision	recall	f1-score	support
Autorizado	0.74	0.90	0.81	30832
Negado	0.60	0.33	0.42	14593
accuracy			0.71	45425
macro avg	0.67	0.61	0.62	45425
weighted avg	0.69	0.71	0.69	45425

Table 1: Tabela de classificação

5 Resultados

Os resultados obtidos no teste foram de média satisfação, sendo que a acurácia obtida foi de 0.7147826086956521 ou 71%, resultando no melhor resultado obtido no treino de 71%. É necessário reavaliar algumas variaveis utilizadas para que se possa obter uma melhor exatidão.

Foi demonstrado a classificação de acurácia na tabela 1.

6 Referências bibliográficas

- https://scikit-learn.org/stable/modules/generated/